

REMARKS

In an Office Action dated March 28, 2006, responsive to a Request for Continued Examination and accompanying Amendment, the Examiner rejected claims 1, 4, 5, and 7-11 under 35 U.S.C. 102(e) as anticipated by Wygodny (US 6,282,701 B1); and rejected claims 2, 12 and 25-32 under 35 U.S.C. 103(a) as unpatentable over *Wygodny* in view of Lindsey (US 5,896,536).

Applicants have amended all independent claims herein to further clarify the scope of the claimed invention. In particular, the independent claims are amended to recite that the trigger expression represents a non-executable data value having a state, and to re-phrase somewhat the operable limitation regarding monitoring the triggering expression. As amended, the claims are patentable over the cited art.

Applicants have discussed the subject references at length in the previous responses filed herein, and applicants incorporate by reference those discussions for a background understanding of the issues involved herein, without necessarily repeating verbatim everything said in response to the previous office actions.

To briefly review the operative differences between applicants' claimed invention and the cited art, applicants provide a tracing technique, in which the *triggering event* which causes data to be collected is a *reference to a state variable data value* occurring during execution of the program being traced.

In a conventional tracing method, a user specifies one or more code statements to serve as triggering events for a trace, and one or more state variables to be traced upon the occurrence of a triggering event. The tracing software then monitors the execution of the program being traced to

detect the occurrence of one of the specified code statements. Encountering a specified code statement during execution constitutes a triggering event, which causes the tracing software to save the state of the user-specified values being traced.

Thus, although conventional tracing software will save the state of any of various values specified by the user, it is not designed to trigger on a reference to one of the values. Rather, it triggers on encountering a particular code statement. In conventional art, the reference to the state variable is therefore not a “triggering event” which causes state data to be saved during a trace, although, once the triggering event occurs, it is data which might be saved. It is this distinction between a triggering event and the data which is saved once the trigger occurs which is critical to the patentability of applicants’ claims herein.

In the most recent office action, the Examiner suggested clarifying the nature of the recited “triggering expression”. Applicants appreciate the suggestion, and have been attempting to do just that, but evidently not everybody perceives the same concept from the same claim language. While the claims appear to applicants to be sufficiently clear in this regard already, applicants have no objection to additional clarification of the claims if that will further prosecution of the present matter.

As applicants understand the Examiner’s rejection, the Examiner is taking the position that a “trigger expression” could be a conventional break point or code point, i.e., a statement in the code. Without necessarily agreeing with this interpretation, applicants have nevertheless amended the independent claims to recite that the trigger expression represents a non-executable data value having a state, in an attempt to clarify that the trigger expression is *not a code statement*.

Applicants’ representative amended claim 1 recites:

A method of tracing the activity of an expression, said method comprising the machine-implemented steps of:

- (a) receiving, from a user, a specification of a machine-implemented process in which a trigger expression is to be traced;
- (b) receiving, from a user, a specification of the trigger expression to be traced in the machine-implemented process, said *trigger expression representing a non-executable data value having a state*;
- (c) responsive to steps (a) and (b), monitoring execution of said machine-implemented process to detect occurrences of a plurality of references to a location in machine memory representing a state of said trigger expression, wherein *each said occurrence of a reference to a location in machine memory representing a state of said trigger expression occurs as a result of executing said machine-implemented process*;
- (d) responsive to each detected occurrence of a reference to said location in machine memory representing a state of said trigger expression, storing the respective state of the trigger expression at the time of the respective detected occurrence of a reference to said location in machine memory representing a state of said trigger expression to create a history of said trigger expression within the machine-implemented process, said storing step being performed without interrupting the machine-implemented process; and
- (e) restoring the state of the trigger expression when requested. [emphasis added]

The remaining independent claims, while not identical in scope, contain analogous limitations to those italicized above.

Wygodny, the primary reference cited herein, discloses a tracing system for remote use, in which trace files can be generated during program execution for later analysis at a remote location. Although Wygodny refers to “tracing” values of state variables, they are referring to the *data that is collected with the trace, not the event which triggers collection of data*. Wygodny is entirely conventional in their triggering event. As per Wygodny, the triggering event is one or more trace points *in the code*. These are specified by the user, and, upon being encountered during execution, the values of the appropriate state variables are saved.

Thus, Wygodny fails to disclose selecting a *trigger expression* representing a *non-executable data value having a state*, and monitoring execution to *detect references (or accesses)*

to a memory location representing the trigger expression, where the references or accesses occur as a result of executing the machine-implemented process or program. If *Wygodny*'s trace points represent "trigger expressions", then the limitation that a trigger expression is a non-executable data value having a state is clearly not met, for trace points are just lines of code. If *Wygodny*'s state variables represent "trigger expressions", then the limitation that execution is monitored to detect references to the memory location of the trigger expression is not met, since the only thing monitored in *Wygodny* is the trace points (code statements). Accordingly, the claims as amended are not anticipated by *Wygodny*.

Nor are the amended claims obvious over *Wygodny*. The thrust of *Wygodny* is the collection of trace data in a remote or distributed environment. Although almost any state data might conceivably be collected, there is simply nothing in *Wygodny* that would suggest collecting state data responsive to a memory access for a triggering expression, where the triggering expression represents a "non-executable data value having a state", i.e. a state variable.

Lindsey, the secondary reference, discloses intercepting messages intended for specific data in an object-oriented program, and capturing trace data depending on the parameters of the message. *Lindsey* is cited to show imposing conditions on the triggering expression. I.e., according to *Lindsey*, conditions can be attached to the collection of trace data from an intercepted message. *Lindsey* does not teach or suggest, alone or in combination with *Wygodny*, that the triggering event which causes state data to be collected is a reference to a memory location corresponding to a triggering expression input by a user, the triggering expression being a state variable.

In view of the foregoing, applicants submit that the claims are now in condition for allowance, and respectfully request reconsideration and allowance of all claims. In addition, the

Examiner is encouraged to contact applicants' attorney by telephone if there are outstanding issues left to be resolved to place this case in condition for allowance.

Respectfully submitted,

CARY L BATES, et al.

A handwritten signature in black ink, appearing to read 'Roy W. Truelson', with a long horizontal flourish extending to the right.

By: _____
Roy W. Truelson
Registration No. 34,265

Telephone: (507) 202-8725